

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2001 年 11 月 1 日 (01.11.2001)

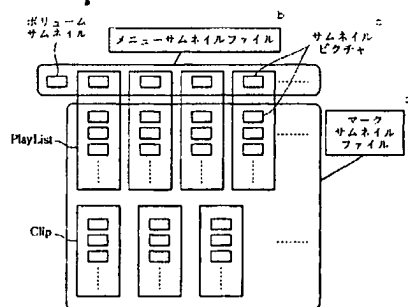
PCT

(10) 国際公開番号
WO 01/82609 A1

- (51) 国際特許分類: H04N 5/93, G11B 20/10 (HAMADA, Toshiya) [JP/JP]. 加藤元樹 (KATO, Motoki) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).
- (21) 国際出願番号: PCT/JP01/03416
- (22) 国際出願日: 2001 年 4 月 20 日 (20.04.2001) (74) 代理人: 小池 晃, 外 (KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (81) 指定国 (国内): CN, KR, US.
- (30) 優先権データ: 特願2000-185479 2000 年 4 月 21 日 (21.04.2000) JP (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP). 添付公開書類:
— 国際調査報告書
- (72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 浜田俊也
- 2 文字コード及び他の略語については、定期発行される各 PCT ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: RECORDING APPARATUS AND METHOD, REPRODUCING APPARATUS AND METHOD, RECORDED MEDIUM, AND PROGRAM

(54) 発明の名称: 記録装置及び方法、再生装置及び方法、記録媒体並びにプログラム



a...VOLUME THUMBNAİL
b...MENU THUMBNAİL FILE
c...THUMBNAİL PICTURE
d...MARK THUMBNAİL FILE

(57) Abstract: A thumbnail image representative of inputted time-varying image data is created as a first thumbnail data set from the time-varying image data. A thumbnail image of an image extracted from the time-varying image as a characteristic image or a thumbnail image of an image specified by the user is created as a second thumbnail data set. The first and second thumbnail data sets are recorded on a recording medium as mutually independent groups. Therefore the user can readily select recorded data by using the recorded image data.

WO 01/82609 A1



(57) 要約:

入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成し、第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するようにしたので、その記録されている画像データを用いることにより、ユーザが記録されているデータを選択を簡便に行える。

明細書

記録装置及び方法、再生装置及び方法、記録媒体並びにプログラム

技術分野

本発明は、データを代表するサムネイルをデータに付加する記録装置及び方法、再生装置及び方法、記録媒体並びにプログラムに関する。

背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group) 2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。或いは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信

号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要がある。

また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

上述したような装置により、複数のデータ（例えば、映像データや音声データ等から構成される番組のデータ）が記録されている記録媒体を再生する際、記録媒体に記録されるデータ量が増加するに従い、どの番組を再生するのか、番組中のどのシーンから再生するのかといった、再生させる前の処理が煩雑になるといった課題があった。

発明の開示

本発明の目的は、このような状況に鑑みて、データを代表するサムネイルをデータに付加することにより、再生させるデータを簡単に選択できるようにすることにある。

本発明に係る記録装置は、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、

ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成手段と、生成手段により生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録する記録手段とを有する。

生成手段は、第1のサムネイルデータと第2のサムネイルデータを、各々独立したファイルとして生成するようにすることができる。

生成手段は、第1のサムネイルデータと第2のサムネイルデータを各々独立したデータブロックとするとともに1つのファイルとして生成するようにすることができる。

生成手段は、第1のサムネイルデータに対応する番号で第1のサムネイルデータを管理する第1の管理データも生成すると共に、第2のサムネイルデータに対応する番号で第2のサムネイルデータを管理する第2の管理データも生成し、記録手段は、第1の管理データ及び第2の管理データを記録媒体に記録するようにすることができる。

第1の管理データと第2の管理データは、管理するサムネイル画像の画像データのフォーマット形式を示すデータを含むようにすることができる。

記録手段は、第1のサムネイルデータ又は第2のサムネイルデータに含まれるサムネイル画像の画像データを、所定の大きさのブロック単位で記録するようにすることができる。

記録手段は、第1のサムネイル画像の参照先を示す情報を更に別ファイルとして記録媒体に記録するようにすることができる。

記録手段は、第2のサムネイルデータに含まれるサムネイル画像の参照先を示す情報を更に記録するようにすることができる。

本発明に係る記録方法は、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するた

めの制御を行う記録制御ステップとを含む。

本発明に係る記録媒体のプログラムは、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含む。

本発明に係るプログラムは、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとをコンピュータに実行させる。

本発明に係る再生装置は、A Vストリームの再生が指示された場合、A Vストリームの内容を示すサムネイル画像の画像データを管理する管理データを読み出す第1の読出手段と、第1の読出手段により読み出された管理データに基づき、画像データを読み出す第2の読出手段と、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出手段と、第3の読出手段により読み出された管理データに基づき、画像データを読み出す第4の読出手段とを含む。

本発明に係る再生方法は、A Vストリームの再生が指示された場合、A Vストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読出制御ステップと、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画

像の画像データと、その画像データを管理する管理データを読み出す第3の読み出しステップと、第3の読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読出制御ステップとを含む。

本発明に係る記録媒体のプログラムは、A Vストリームの再生が指示された場合、A Vストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読出制御ステップと、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出制御ステップと、第3の読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読出制御ステップとを含む。

本発明に係るプログラムは、A Vストリームの再生が指示された場合、A Vストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読出制御ステップと、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出制御ステップと、第3の読出制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読出制御ステップとを含む。

本発明に係る記録媒体は、A Vストリーム、A Vストリームから、A Vストリームの内容を示すサムネイル画像の画像データと、そのサムネイル画像のデータを管理する管理データから構成される第1のデータ、及び、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザが指定した画面のサムネイル画像の画像データと、その画像データを管理する管理データから構成される第2のデータが記録されている。

本発明に係る記録装置及び方法、並びに第1のプログラムにおいては、入力さ

れた動画像データから、当該動画像データを代表するサムネイル画像が第1のサムネイルデータとして生成されると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像が第2のサムネイルデータとして生成され、第1のサムネイルデータと第2のサムネイルデータが、各々独立したグループとして記録媒体に記録される。

本発明に係る再生装置及び方法、並びに第2のプログラムにおいては、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データが読み出され、読み出された管理データに基づき、画像データが読み出され、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データが読み出され、読み出された管理データに基づき、画像データが読み出される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の構成を示す図である。

図2は、ClipとPlaylistの関係を表す図である。

図3は、AVストリームを管理する構造のUML図である。

図4は、DVRシステムのディレクトリ構成を表す図である。

図5は、info.dvrを説明する図である。

図6は、DVRVolume()を説明する図である。

図7は、UIAppInfoVolume()を説明する図である。

図8は、TableOfPlayLists()を説明する図である。

図9は、UIAppInfoPlayList()を説明する図である。

図10は、zzzzz.clpiを説明する図である。

図11は、ClipMark()を説明する図である。

図12は、マークを説明する図である。

図 1 3 は、xxxxx.rpls、yyyyy.vplsを説明する図である。

図 1 4 は、Playlist()を説明する図である。

図 1 5 は、PlayItem()を説明する図である。

図 1 6 は、PlaylistMark()を説明する図である。

図 1 7 は、メニューサムネイルを説明する図である。

図 1 8 は、プレイリストに付けられるマークを説明する図である。

図 1 9 は、クリップに付けられるマークを説明する図である。

図 2 0 は、サムネイルを格納するファイルを説明する図である。

図 2 1 は、menu_thmb/mark.thmbを説明する図である。

図 2 2 は、Thumbnail()を説明する図である。

図 2 3 は、Thumbnail_picture_formatを説明する図である。

図 2 4 A 及び図 2 4 B は、サムネイルの画像データをtn_block()に格納する方法を説明する図である。

図 2 5 は、他のディレクトリ・ファイル構造を示す図である。

図 2 6 は、図 2 5 に示したファイル構造に対応するmenu_thmb/mark.thmbを説明する図である。

図 2 7 は、図 2 5 に示したファイル構造に対応するTableOfPlayLists()を説明する図である。

図 2 8 は、図 2 5 に示したファイル構造に対応するPlaylist()を説明する図である。

図 2 9 は、図 2 5 に示したファイル構造に対応するUIAppInfoPlayList()を説明する図である。

図 3 0 は、マークサムネイルの作成手順を表したフローチャートである。

図 3 1 は、メニューサムネイルの作成手順を表したフローチャートである。

図 3 2 は、DVRシステムの他のディレクトリ構成を表す図である。

図 3 3 は、サムネイルのヘッダ情報ファイルのシンタクスを示す図である。

図 3 4 は、display_aspect_ratioを説明する図である。

図 3 5 は、color_spaceを説明する図である。

図 3 6 は、サムネイルのピクチャデータファイルのシンタクスを示す図である。

図 3 7 は、tn_blockへのデータの格納について説明する図である。

図 3 8 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された記録装置及び方法、再生装置及び方法、記録媒体並びにプログラムについて、図面を参照して説明する。図 1 は、本発明を適用した記録再生装置 1 の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置 1 は、アナログデータ、又は、デジタルデータを入力し、記録することができる。

端子 1 1 には、アナログのビデオ信号が、端子 1 2 には、アナログのオーディオ信号が、それぞれ入力される。端子 1 1 に入力されたビデオ信号は、解析部 1 4 と A V エンコーダ 1 5 に、それぞれ出力される。端子 1 2 に入力されたオーディオ信号は、A V エンコーダ 1 5 にのみ出力される。解析部 1 4 は、入力されたビデオ信号からシーンチェンジ等の特徴点を抽出する。

A V エンコーダ 1 5 は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム (V)、符号化オーディオストリーム (A)、及び A V 同期等のシステム情報 (S) をマルチプレクサ 1 6 に出力する。

符号化ビデオストリームは、例えば、M P E G (Moving Picture Expert Group) 2 方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、M P E G 1 方式により符号化されたオーディオストリームや、ドルビー A C 3 方式により符号化されたオーディオストリーム等である。マルチプレクサ 1 6 は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ 1 7 を介して多重化ストリーム解析部 1 8 とソースパケットイザ 1 9 に出力する。

多重化ストリームは、例えば、M P E G 2 トランスポートストリームや M P E G 2 プログラムストリームである。ソースパケットイザ 1 9 は、入力された多重化ストリームを、そのストリームを記録させる記録媒体 1 0 0 のアプリケーションフォーマットに従って、ソースパケットから構成される A V ストリームを符号

化する。AVストリームは、ECC（誤り訂正）符号化部20、変調部21で所定の処理が施され、書込部22に出力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む（記録する）。

デジタルインタフェース又はデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、それらは、トランスベアレントに記録する方式と、記録ビットレートを下げ等々の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子24から制御部23へ入力される。

入力トランスポートストリームをトランスベアレントに記録する場合、端子13に入力されたトランスポートストリームは、多重化ストリーム解析部18と、ソースパケッタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム（V）、オーディオストリーム（A）、及びシステム情報（S）を抽出する。

デマルチプレクサ26により抽出されたストリーム（情報）の内、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム（V）をマルチプレクサ16に出力する。

一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、及び、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ス

トリームとして多重化ストリーム解析部 18 とソースパケットタイザ 19 にスイッチ 17 を介して出力される。これ以後の記録媒体 100 へ AV ストリームが記録されるまでの処理は、上述の入力オーディオ信号と、ビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

本例の記録再生装置 1 は、AV ストリームのファイルを記録媒体 100 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部 23 により作成される。制御部 23 への入力情報は、解析部 14 からの動画像の特徴情報、多重化ストリーム解析部 18 からの AV ストリームの特徴情報、及び端子 24 から入力されるユーザからの指示情報である。

解析部 14 から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関係する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル (CM) の開始・終了点等の指定情報 (マーク) であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

多重化ストリーム解析部 18 からの AV ストリームの特徴情報は、記録される AV ストリームの符号化情報に関係する情報であり、例えば、AV ストリーム内の I ピクチャのアドレス情報、AV ストリームの符号化パラメータ、AV ストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関係する情報 (マーク) 等である。

端子 24 からのユーザの指示情報は、AV ストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報等である。

制御部 23 は、上記の入力情報に基づいて、AV ストリームのデータベース (Clip)、AV ストリームの再生区間 (PlayItem) をグループ化したもの (PlayList) のデータベース、記録媒体 100 の記録内容の管理情報 (info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AV ストリームと同様にして、ECC 符号化部 20、変調部 21 で処理されて、書込部 22 へ入力される。書込部 22 は、制御部 23 から出力される制御信号に基づいて、記録媒体 100 へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生される場合、先ず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29、ECC復号部30の処理を経て、制御部23へ入力される。

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読出部28に指示する。読出部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、更にECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26に出力できるストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、及びAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読出部28に指示する。

例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読出部28に指示する。

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読出部28に指示する。

読出部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書込部22に指示する。

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読出部28にイン点側のピ

クチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読出部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタイザ31を経て、デマルチプレクサ26に出力される。

制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法（picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て）と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム（V）、オーディオストリーム（A）、及びシステム情報（S）に分離する。ビデオストリームは、「AVデコーダ27に入力されるデータ」と「マルチプレクサ16に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書込部22に入力される。書込部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

このような記録再生装置1において記録媒体100に記録されるデータ（記録媒体100から再生されるデータ）について説明する。MPEG Video、MPEG Audio等の符号化方式で符号化され、MPEG-2 Systemsに従って多重化されたビットストリームを、ファイルシステムが扱うファイルの形にしてディスク（以下、適宜、記録媒体100をディスク状の記録媒体とし、ディスクという）に記録したものをAV(Audio Video)ストリームファイル（又はClip AVストリーム）という。

このようなAVストリームファイルの一部又は全部の範囲を指定して、必要な部分だけを並べて再生する再生順序指定の仕組みを説明する。図2のように、AVストリームファイルの一部又は全部の範囲を指定して、必要な部分だけを再生する再生順序指定がPlaylistである。Playlistは、ユーザから見て、ひとまとまりの単位である。最も簡単な構成になるのは、記録開始をしてから記録終了までの単位で、編集をしなければ、これが1つのPlaylistになる。

Playlistは、どのAVストリームを再生するかという、AVストリームファイルの指定と、そのファイル中の再生開始点（イン点）と再生終了点（アウト点）の集まりで構成される。AVストリームファイル、再生開始点、再生終了点等を1組とし、これをPlayitemという。すなわち、PlaylistはPlayitemの集合である。

図2に示したように、Playitemは、あるAVストリームファイルの、イン点、アウト点で指定する範囲を参照する。Playitemを再生するということは、そのPlayitemが参照するAVストリームの一部分を再生することになる。

AVストリームは、MPEG-TSの形等にマルチプレクスされているビットストリームであるが、このAVストリームが記録されるファイルとは別のファイルに、そのビットストリームに対して1対1に対応する情報（クリップ情報：Clip information）を保持する。これは、再生、編集をより容易にするために設けられている。このようなクリップ情報と、AVストリームの両方をひとまとまり（オブジェクト）とみなし、これをClip（クリップ）と呼ぶ。すなわち、ClipはAVストリームと、それに付随する情報から構成される1つのオブジェクトである。

以上のような、関係をUML図で表すと、図3のようになる。図3のUML図で表されるAVストリームファイル、Clip、Playitem、Playlistの構造により、AVストリームファイルを変更しない、任意の部分だけを再生する非破壊再生順序指定が可能になる。

次に、本発明を実現するための、各種の情報が記録又は再生される記録媒体（メディア）上のファイル配置について説明する。メディア上には、図4に示すように、info.dvr、menu.thmb（mark.thmb）、#####.rpls（#####.vpls）（#####は任意の番号）、%%%%.clpi（%%%%は任意の番号）、及び %%%.m2ts（%%%%は、

各m2tsファイルがcpliファイルと1対1に対応するような番号)の、5種類のファイルが記録される。

ディスク上にディレクトリ/DVRを用意し、このディレクトリの下が、1つのディスク記録再生システムで管理される範囲とする。/DVRは、ディスクのルートディレクトリにあってもよいし、所定のディレクトリの下に存在していてもよい。/DVRディレクトリには、info.dvr、 menu.thmb、 mark.thmbというファイルが置かれる。また、/DVRの下には、/PLAYLIST、 /CLIPINF、 /M2TSというディレクトリが置かれる。ファイル####.rp1s、 ####.vp1sは、/PLAYLISTの下に置かれ、/CLIPINFには%%%%.clpiが置かれ、/M2TSには、%%%%.m2tsが置かれる。

ファイルinfo.dvrは、/DVRの下にただ1つある。info.dvrの構造は、図5に示したようなシンタクスで表される。ファイルの内部は、機能別の情報毎にブロックを構成しており、volumeに関する情報はDVRVolume()に、 Playlistの並びに関する情報はTableOfPlayLists()に、記録再生装置1のメーカー固有の情報はMakerPrivateData()に、それぞれ格納される。ファイルの先頭部分には、それらのブロックの先頭を表すアドレスが記述されている。例えば、TableOfPlayLists_Start_addressは、TableOfPlayLists()が開始する位置を、ファイル内での相対バイト数で表したものになっている。

DVRVolume()の構造は図6に示したようなシンタクスで表される。version_numberは、DVRVolume()のバージョン番号を表し、lengthは、length直後のフィールドからDVRVolume()の最後までをバイトで表したものである。ResumeVolume()はresumeに関する情報を格納し、UIAppInfoVolume()はVolumeの属性情報を格納する領域である。

図7は、UIAppInfoVolume()のシンタクスを表したものである。character_setは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法としては、ASCII、Unicode等を選択できる。name_lengthは、Volume_nameフィールドの中に示されるボリューム（ディスク）名のバイト長を示す。Volume_nameは、ボリューム（ディスク）の名前を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリューム（ディスク）の名前を示す。

Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。Volume_protect_flagは、そのボリュームの中のコンテンツをユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（図7中のPIN）を入力できた場合にだけ、ユーザは、そのボリュームのコンテンツを視聴することができる。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザが、そのボリュームのコンテンツを視聴することができる。最初に、ユーザが、ディスクをプレーヤ（記録再生装置1）へ挿入したときにおいて、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、プレーヤは、そのセットされたディスクの中のPlaylistの一覧を表示させる。

以上は、Volumeに対しての再生制限であるが、それぞれのPlaylistの再生制限については、Volume_protect_flagとは無関係であり、それは後述する図9のUIAppInfoPlaylist()の中で定義されるplayback_control_flagによって示される。PINは、4個の0乃至9までの数字であり、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexは、Volumeを代表するサムネイルが存在する場合、そのサムネイルを特定するための番号を格納する領域である。ref_thumbnail_indexで指定されるthumbnail_indexを持つ、ファイルmenu.thmb中のサムネイルが、Volumeを代表するメニューサムネイル（本発明では、VolumeやPlaylistを代表するサムネイルを特にメニューサムネイルという）となる。

rp_info_valid_flagは、これが1である場合に次に続くrp_ref_to_PlayList_file_name, rp_ref_to_PlayItem_id及びrp_time_stampが有効な値を持つこと示す。rp_ref_to_PlayList_file_nameは、上記のVolumeを代表するメニューサムネイルが、所定のPlaylist中の画像から抜き出された画像から作られていることを示し、そのPlaylistファイルの名前を示す。

rp_ref_to_PlayItem_idは、rp_ref_to_PlayList_file_nameで示されるPlaylistの中の1つのPlayItemを指すPlayItem_idを示し、上記のVolumeを代表するメニューサムネイルが、そのPlayItem中の画像から抜き出された画像から作られていることを示す。rp_time_stampは、rp_ref_to_PlayItem_idが指すPlayItem中の1

つの画像のプレゼンテーションタイムスタンプを示し、その画像から上記のVolumeを代表するメニューサムネイルが作られていることを示す。

図8は、TableOfPlayLists()のシンタクスを表したものである。ここでnumber_of_PlayListsはVolume中のPlaylistの数を表し、Playlist_file_nameは####.rpls、####.vpls等のファイル名を指定するものである。UIAppInfoPlaylist()には、Playlistの各種属性がかかれており、シンタクスは図9のようになっている。図9に示したUIAppInfoPlaylist()には、Playlistの再生には直接必要ではない、Playlistの名前、記録日時、記録時間、消去禁止の有無等の、各種属性情報が格納される。その中のref_thumbnail_indexにより、Playlistの代表画としてのサムネイルを指定することができる。すなわち、ref_thumbnail_indexで指定されるthumbnail_indexを持つ、ファイルmenu.thmb中のサムネイルが、このPlaylistを代表するメニューサムネイルとなる。

rp_info_valid_flagは、これが1である場合に、次に続くrp_ref_to_PlayItem_id及びrp_time_stampが有効な値を持つこと示す。rp_ref_to_PlayItem_idは、Playlistの中の1つのPlayItemを指すPlayItem_idを示し、Playlistを代表するメニューサムネイルが、そのPlayItem中の画像から抜き出された画像から作られていることを示す。rp_time_stampは、rp_ref_to_PlayItem_idが指すPlayItem中の1つの画像のプレゼンテーションタイムスタンプを示し、その画像からPlaylistを代表するメニューサムネイルが作られていることを示す。

図4に示したファイル%%%%.clpiは、/CLIPINFの下に、各AVストリームファイル%%%%.m2tsに対応して1つ作られる。%%%%.clpiの構造は、図10に示すようになっている。ファイルの内部は、機能別の情報毎にブロックを構成しており、Clipに関する情報はClipInfo()に、MPEG-2 systemsにおけるトランスポートストリームの時刻基準を表すPCR（プログラムクロックリファレンス）の不連続点に関する情報はSTC_Info()に、MPEG-2 systemsのprogram（プログラム）に関する情報はProgramInfo()に、AVストリーム中のランダムアクセス開始可能点等の特徴的な点を表すCPI（Characteristic Point Information）に関する情報はCPI()に、Clipに付けられた、頭出しのためのインデックス点やコマーシャルの開始・終了点等のマーク情報はClipMark()に、それぞれ格納される。ファイルの先

頭部分には、それらのブロックの先頭を表すアドレスが記述されている。

ここでは、サムネイルに関係するブロックだけを説明する。本例においては、クリップにサムネイルを付ける操作は、クリップのマークにサムネイルを付与することで実現する。クリップに付けられるマークの情報は、ClipMark()に格納されている。図11に、ClipMark()のシンタクスを示す。シンタクス中のmark_typeが、resume、bookmark、skip等のマークの種類を表し、mark_time_stampでマークが付けられた時刻を表す。サムネイルに関係するフィールドは、ref_thumbnail_indexである。ref_thumbnail_indexは、ここでサムネイルの番号を指定することで、サムネイルのデータが格納されているmark.thmbファイル中のサムネイルを特定する。ref_thumbnail_indexで指定されるサムネイルが、このマークに付与されたマークサムネイル（本発明では、マークに付与されたサムネイルを特にマークサムネイルという）となる。

マークは、主としてClip及びPlaylistの中のハイライトや特徴的なシーンを指し示すためにある。また、マークより後の部分は再生を飛ばして、次のPlaylistの再生をするように指示するスキップ機能も、マークによって実現できる。図12は、マークがClip、Playlistに付けられている様子を図で示したものである。以下にマークの特徴をまとめる。

Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーン、例えば、シーンチェンジ点等を指定する。Playlistを再生するときには、そのPlaylistが参照しているClip中のマークを利用してランダムアクセス等ができる。図12では、Clipに対してcommercial (CM) とsceneという、2種類のマークが付けられているが、commercialのマークは左のPlaylistから、sceneマークは二つのPlaylistから利用されている。Playlistに付加されるマークは、主としてユーザによって設定される。例えば、ブックマークやリジューム点等である。図12では、bookmarkとresumeマークが相当する。

Clip又はPlaylistにマークを設定することは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中からそのマークのタイムスタンプを除去することである。それゆえ、マークの設定や削除は、AVストリームを何も変化させない。

次に、図4に示したファイル#####.rpls、#####.vplsについて説明すると、ファイル#####.rpls、#####.vplsは、/PLAYLISTの下に、各playlistに対してどちらか1つ作られる。#####.rpls、#####.vplsの構造は、図13に示すようになっている。ファイルの内部は、機能別の情報毎にブロックを構成しており、Playlistに関する情報はPlaylist()に、Playlistに付けられるマークの情報はPlaylistMark()に、このPlaylistファイルを記録した記録再生装置のメーカ固有の情報はMakerPrivateData()に、それぞれ格納される。ファイルの先頭部分には、ブロックの先頭を表すアドレス(PlaylistMark_Start_address等)が記述されている。これにより、ブロックの前或いは後ろにpadding_byteを挿入することができる。ただし、Playlist()の開始位置は固定で、ファイルの先頭から、例えば、256バイト目と設定されている。

ブロックPlaylist()の内容は、図14のようになっている。最初にversion_numberがあり、以下に続く情報のバージョン番号を表す。lengthは、lengthの直後のフィールドから、Playlist()の終わりまでのバイト長を表す。Playlist_typeは、このPlaylistの種類を表し、CPI_typeは、このPlaylistが持つCPIの種類を表す。number_of_PlayItemsは、このPlaylistを構成するPlayitemの数を表す。number_of_SubPlayItemsは、このPlaylistに付けられているアフレコオーディオ用のPlayitem(SubPlayitem)の数を表す。PlayItem()は、Playitemの情報を、SubPlayItem()は、SubPlayitemの情報を格納する。

ブロックPlayItem()の内容は、図15のようになっている。Clip_Information_file_nameは、このPlayItemが参照しているClip情報ファイル(拡張子がclpiであるファイル)のファイル名が文字列で格納されている。STC_sequence_idは、program中に存在する、PCRが連続な時間範囲の区間を表す。この区間内では、一貫した連続時間軸が定義できるようになっているので、PlayItemの開始・終了点を一意に定めることができる。つまり、各PlayItemの開始点と終了点は、同一のSTC_sequenceに存在していなければならない。

IN_timeは、このPlayItemの開始点の、STC_sequence 上でのpts (Presentation Time Stamp)を表し、OUT_time はPlayItemの終了点の、STC_sequence 上でのptsを表す。connection_conditionは、このPlayitemが次のPlayitemとの間でどの

ような接続がされているかを表す情報であり、Playitemの間を継ぎ目なく再生できるかどうかの条件を表す。

BridgeSequenceInfo()は、Playitemの継ぎ目の部分で、本来再生すべきビットストリームとは異なるビットストリームに飛び、それを代わりに再生することで、Playitemの間をシームレスに再生する機能を実現する際に作成されるビットストリーム（ブリッジシーケンス）に関する情報を格納する。program_numberは、このPlayitemが参照しているprogram（MPEG Systemsで定義されている、ビデオ・オーディオ等のエレメンタリストリームのまとまりをいう。いわゆるテレビジョン放送のチャンネルに相当するものである）のprogram_numberを表す。

以上が、Playlist、Playitemを構成するデータ構造の概要である。このようなデータ構造により、AVストリーム中の再生したい部分をIN、OUT点の組で指定したPlayitemの並びでPlaylistを構築し、ユーザが認識するひとまとまりの再生単位を管理することが可能になる。

本発明の例では、プレイリスト上の任意の時刻にサムネイルを付ける操作は、プレイリストのマークにサムネイルを付与することで実現している。Playlistに付けられるマークの情報は、PlayListMark()に格納されている。図16は、PlayListMark()のシンタクスを示す図である。シンタクス中のmark_typeが、resume、bookmark、skip等のマークの種類を表し、mark_time_stampでマークが付けられた時刻を表す。ref_thumbnail_indexは、ここでサムネイルの番号を指定することで、サムネイルのデータが格納されているmark.thmbファイル中のサムネイルを特定する。ref_thumbnail_indexで指定されるサムネイルが、このマークに対応するマークサムネイルとなる。

次に、サムネイルの詳細について説明する。サムネイルとは、Volume、Playlist、Clipに付随する静止画のことを指す。サムネイルには2種類ある。1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソルを操作して見たいものを選択するためのメニュー画面で使われる。もう一つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画とは、ディスクをプレーヤに入れたときに、ディスクの内容を表す静

止画を最初に表示する場合等に用いることを想定している。Playlistの代表画とは、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

Playlistの代表画の最も簡単な実現方法は、Playlistの最初の画像をサムネイルにすることであるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像を決めることができるようにしておく。以上2種類のサムネイルをメニューサムネイルという。メニューサムネイルは、頻繁に表示されるため、ディスクから高速に読み出されることが可能である必要がある。この要求を満たすには、全てのメニューサムネイルを1つのファイルに格納することが効率的である。必ずしもボリューム内の動画から抜き出したピクチャだけではなく、図17に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込んだ画像でもよい。

一方、ClipとPlaylistは複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要がある。このようなマーク点を表すピクチャをマークサムネイルという。よって、サムネイルの元となるものは、図18や図19に示したように、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。メニューサムネイルと異なり、マークサムネイルはPlaylistの詳細を表すときに使われるサブメニュー等で使われるため、短いアクセス時間で読み出されなくてもよい。

そのため、サムネイルが必要になる度に、プレーヤがファイルを開き、ファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを持つことはしない。図20は、以上の関係を示した図である。

サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に、かつ、高速に実行されなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つ

のtn_block()に格納される。1つの画像データは、連続したtn_block()に格納される。tn_block()の列には、使用されていないtn_block()が存在してもよい。1つのサムネイル画像のバイト長は可変である。

図21は、サムネイルのデータを格納するファイルのシンタクスである。このファイルには、Thumbnail()がただ一つ存在する。図22は、thumbnail()のシンタクスを表す。version_numberは、このthumbnail()のバージョンナンバーを示す4個のキャラクター文字を表す。lengthは、このlengthフィールドの直後からthumbnail()の最後までのthumbnail()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、thumbnail()中の最初のtn_block()の、thumbnail()の先頭からの開始バイトアドレスを示す32ビットの符号なし整数である。

number_of_thumbnailsは、このthumbnail()に格納されているサムネイル画像の数を示す16ビットの符号なし整数である。tn_block_sizeは、1 tn_block()の大きさをキロバイト単位で表す16ビットの符号なし整数である。例えば、tn_block_size が1であるのは、1つのtn_block()のサイズが1024バイトであることを表す。number_of_tn_blocksは、この thumbnail()中のtn_block()の数を表す16ビットの符号なし整数である。

thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ1回分のサムネイル情報で表されるサムネイルのインデクス番号を表す16ビットの符号なし整数である。 thumbnail_index として、0xFFFF という値を使用してはならない。thumbnail_index はref_thumbnail_indexによって参照される。thumbnail_picture_format は、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図23に示したような値をとる。図23中のDCF (Design rule for Camera File System) とPNG (Portable Network Graphics) は”menu.thmb” 内でのみ許される。すなわちマークサムネイルは、値”0x00” (MPEG-2 Video I-picture)をとらなければならない。

picture_data_size は、符号化されたサムネイル画像のバイト長を表す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_block()のtn_block_numberを表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tn_blockの先頭と一致していなければならない。

`start_tn_block_number` は、0 から始まり、`tn_block` の for-ループ中の変数 `k` の値に関係している。`x_picture_length` は、サムネイル画像の水平方向のピクセル数を表す 16 ビットの符号なし整数である。

`y_picture_length` は、サムネイル画像の垂直方向のピクセル数を表す 16 ビットの符号なし整数である。`tn_block()` は、ピクチャデータを格納する領域である。`thumbnail()` 中の全ての `tn_block()` は、`tn_block_size` で定められる同一の大きさを有していなければならない。図 24 A 及び図 24 B は、画像データがどのように `tn_block()` に格納されるかを模式的に表した図である。図 24 A 及び図 24 B に示したように、各画像データは `tn_block()` の先頭から始まり、1 `tn_block()` を超える大きさの場合は、連続する次の `tn_block()` を使用することにより格納される。

画像データは、可変長であるが、1 `tn_block()` は、固定長である。このように、可変長であるデータを固定長のデータに変換して扱うようにすることにより、新たな画像データの追加や、画像データの削除といった処理に対してブロック単位で対処できるため、アドレスの管理等、簡便に行うことが可能となる。

ここで、サムネイル記録についてのディレクトリ・ファイル構造、シンタクスの別案を示す。まず、図 4 で表されるディレクトリ・ファイル構造の別案として、図 25 の構造が考えられる。図 4 では、マークサムネイルを記録するためのファイルが /DVR の下に 1 つ、`mark.thmb` だけであるのに対し、図 25 では、プレイリスト、クリップ毎に 1 対 1 に対応してファイル `xxxxxx.thmb` (`xxxxxx` は対応するプレイリスト又はクリップを表す、拡張子を除いたファイル名) が作られている。

例えば、`#####.rpls` で表されるプレイリストが有するマークサムネイルの画像は、同じディレクトリ内の `#####.thmb` に格納されることになる。同様に、`%%%%.clpi` で表されるクリップが有するマークサムネイルの画像は、同じディレクトリ内の `%%%%.thmb` に格納されることになる。プレイリスト・クリップのマークサムネイルは、ファイルの拡張子を除いた名前の部分が同一の `.thmb` ファイルに記録されるという制約があるだけで、`xxxxxx.thmb` 内のシンタクスは、図 26 のように、`menu.thmb` や `mark.thmb` (図 21) と同一である。

また、プレイリストのメニューサムネイルについて、図 8、図 9 では、サムネイルを特定する `ref_thumbnail_index` を、ファイル `Info.dvr` の `TableOfPlayLists`

()の中のUIAppInfoPlayList()に置いているが、UIAppInfoPlayList()をInfo.dvrではなく、各プレイリストファイル中に入れる方法もある。このような場合、UIAppInfoPlayList()を、TableOfPlayLists()から、プレイリストファイル中のplaylist()へ移すことになり、図8に示したシンタクスは図27に示すシンタクスのようになり、図14に示したシンタクスは図28に示すシンタクスのようになる。図28中のUIAppInfoPlayList()は、図29に示すシンタクスのようになり、ここにref_thumbnail_indexが入る。

以上のようなファイル、シンタクスでもサムネイルの記録が可能である。

次に、図30に示したフローチャートを参照して、マークサムネイルの作成について説明する。ステップS1において、ユーザは、再生したいPlayList(プレイリスト)を選択する。ステップS2において、選択されたPlayListに基づいて、AVストリームの再生が開始される。ステップS3において、ユーザは、再生されているAVストリームを視聴し、マークしたいシーンを探索する。ユーザは、マークしたいシーンが探索できた場合、記録再生装置1に付属するリモートコントローラ(不図示)のマークボタンを操作する。この操作は、再生中に操作されても、一時停止された状態で操作されてもよい。

ステップS3において、ユーザによりマークボタンが操作されると、ステップS4において、ユーザの指示に対応する処理としてマーク位置が決定される。マーク位置が決定されると、ステップS5において、サムネイル画像にする画像の選択が行われる。マーク位置が指示された時点で、制御部23(図1)は、サムネイルを作成するかどうかをユーザに尋ね、作成する場合には、サムネイルとする画像の候補をコマ送り等によりユーザに提供し、ユーザがサムネイル画像を選択するようにしてもよい。

ユーザ、又は、制御部23により、サムネイル画像にする画像が選択されると、制御部23は、ステップS6において、AVデコーダ15から画像を取り込み制御部23のRAM(Random Access Memory)(不図示)に転送する。ステップS7において、画像を圧縮するか否かが判断され、画像を圧縮すると判断された場合、制御部23は、ステップS8に進み、画像圧縮を行い、再びデータをRAMに戻す。ステップS7において、画像を圧縮しないと判断された場合、ステップ

S 8 の処理はスキップされ、ステップ S 9 に進む。

ステップ S 9 において、制御部 2 3 は、サムネイルの符号化方式、サムネイルを識別する thumbnail_id、サイズ（バイト数）、X、Y 方向のピクセル数から構成されるサムネイルのヘッダ情報を作成し、画像データを tn_block() の単位に分割する。ステップ S 1 0 において、E C C 符号化ブロック 2 0 にヘッダ情報と tn_block() の形に分割した画像データが転送され、書込部 2 2 を経て、mark.thmb ファイルとして記録媒体 1 0 0 に書き込まれる。

次に、メニューサムネイルを作成について、図 3 1 に示したフローチャートを参照して説明する。メニューサムネイルはボリューム又は各プレイリストに対して付けられるので、制御部 2 3 は、ステップ S 2 1 において、ユーザに対して、メニューサムネイルを付ける対象を選択させる。ステップ S 2 2 において、制御部 2 3 は、サムネイルを外部から取り込むか、プレイリスト中のあるシーンから取り込むかをユーザに選択させる。

ステップ S 2 2 において、サムネイルを外部から取り込むと判断された場合、ステップ S 2 3 に進み、ユーザが指定した入力端子或いはファイルから画像データが取り込まれる。取り込みが終了されると、ステップ S 2 6 に進む。

一方、ステップ S 2 2 において、サムネイルを外部から取り込むのではないと判断された場合、ステップ S 2 4 に進み、制御部 2 3 は、先ず、ユーザに対して、どのプレイリストからサムネイル画像を取得するのかが選択させ、その後、選択されたプレイリストの再生を開始する。そして、ステップ S 2 5 において、ユーザにサムネイルにしたいシーンを選択させる。

ステップ S 2 3、或いは、ステップ S 2 5 において、サムネイル画像が選択されると、ステップ S 2 6 に進む。ステップ S 2 6 以降の処理は、図 3 0 に示したフローチャートのステップ S 5 以降の処理と同様であるので、その説明は省略する。

このように、サムネイル画像をプレイリストに付与することにより、それらのサムネイル画像を管理しやすくなり、記録されている A V ストリームを再生する際にも、再生させる A V ストリームの選択等の処理を簡便に行わせることが可能となる。

図32は、ディスク上に作られるディレクトリ及びファイルの別の例を示す図である。"menu.tidx"と"menu.tdat"は、メニューサムネイル、すなわちVolumeを代表する1つのピクチャ及びPlayList毎に、それを代表する1つのピクチャの情報をストアする。全てのメニューサムネイルのヘッダ情報は、1つのmenu.tidxに集めて管理される。全てのメニューサムネイルのピクチャデータは、1つのmenu.tdatに集められて管理される。

"mark.tidx"と"mark.tdat"は、マークサムネイル、すなわちマーク点で指されるピクチャについての情報をストアする。Volume中の全てのClip及びPlayListに付加されている、全てのマークサムネイルのヘッダ情報は、1つのmark.tidxに集めて管理される。全てのマークサムネイルのピクチャデータは、1つのmark.tdatに集めて管理される。

すなわち、このファイル構造では、サムネイルのヘッダ情報とサムネイルのピクチャデータが、別々のファイルに分けて管理される。この理由は、ファイルシステムが行うファイルの管理方法に関係する。すなわち、サムネイルのヘッダ情報のファイルは、比較的重要な情報であるため、ファイルシステムがディスク上にデータを2重書きして管理する。

これは、例えば、片方のデータがディスクの傷等によって失われてしまったような場合に対応できるようにするためであり、データのバックアップの意味合いを持つ。ピクチャデータのファイルは、2重書きはしないが、その理由は、ピクチャデータは比較的数据量が大きくなり、これを2重書きするとディスク上の必要容量が大きくなるためである。

menu.tdat及びmark.tdatにストアされるピクチャデータの符号化方法、サンプリング構造、及びスキャン構造は、ピクチャ毎に符号化方法を変更することが可能であるが、全てのピクチャデータで同じであるほうが、記録再生装置1の構成を簡単化できる。例えば、J F I F (JPEG File Interchange Format)、コンポーネント信号、プログレッシブスキャンフォーマットが使用される。

これら4個のファイルのシンタクスとセマンティクスを説明する。"menu.tidx"と"mark.tidx"は、同じシンタクス構造を持つ。図33は、"menu.tidx"と"mark.tidx"のシンタクス構造を示す。version_numberは、このサムネイルヘッダ情報フ

ファイルのバージョンナンバーを示す4個の数字である。

lengthは、このlengthフィールドの直後のバイトからmenu.tidx/mark.tidxの最後のバイトまでのバイト数である。number_of_thumbnailsは、menu.tidxの場合にはmenu.tdatにストアされているサムネイルピクチャの数であり、mark.tidxの場合、mark.tdatにストアされているサムネイルピクチャの数である。

tn_block_sizeは、menu.tidxの場合、menu.tdatの中の1つのtn_blockのサイズを示し、mark.tidxの場合、mark.tdatの中の1つのtn_blockのサイズを示す。このサイズは、1024バイトを単位とする大きさである。例えば、tn_block_size=1は、1つのtn_blockのサイズが1024バイトであることを示す。1つのサムネイルピクチャは、1つのtn_blockの中にストアされなければならない。

number_of_tn_blocksは、menu.tidxの場合、menu.tdatの中にあるtn_blockの数を示し、mark.tidxの場合にはmark.tdatの中にあるtn_blockの数を示す。thumbnail_indexは、このthumbnail_indexフィールドに続くサムネイル情報のインデックス番号を表す。thumbnail_indexとして、0xFFFF という値を使用してはならない。

menu.tidxの場合、thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlayList()の中のref_thumbnail_indexによって参照される。mark.tidxの場合、thumbnail_indexはPlayListMark()及びClipMark()の中のref_thumbnail_indexによって参照される。

ref_to_tn_block_idは、menu.tidxの場合、menu.tdat中の1つのtn_blockを示し、そのtn_blockは、thumbnail_indexで指されるピクチャデータをストアしている。ref_to_tn_block_idの値は、menu.tdatのシンタクス中のtn_block_idの値を参照する。

mark.tidxの場合、mark.tdat中の1つのtn_blockを示し、そのtn_blockは、thumbnail_indexで指されるピクチャデータをストアしている。ref_to_tn_block_idの値は、menu.tdatのシンタクス中のtn_block_idの値を参照する。

picture_byte_sizeは、thumbnail_indexで指される1つの符号化サムネイルピクチャのデータ長をバイト単位で示す。picture_byte_sizeは、1024*tn_block_sizeの値以下でなければならない。すなわち、記録再生装置1は、1つの符号化サムネイルピクチャのデータ長を1024*tn_block_sizeの値以下になるように、符号

化しなければならない。

horizontal_picture_sizeは、thumbnail_indexで指される符号化サムネイルピクチャの水平方向の画素数を示す。vertical_picture_sizeは、thumbnail_indexで指される符号化サムネイルピクチャの垂直方向の画素数を示す。display_aspect_ratioは、thumbnail_indexで指される符号化サムネイルピクチャのディスプレイ・アスペクト・レシオを示す。値の意味を図34に示す。

color_spaceは、Y,Cb,Crのコンポーネント信号をR,G,Bのコンポーネント信号へ変換するときのフォーマットを示す。値の意味を図35に示す。

"menu.tdat"と"mark.tdat"は、同じシンタクス構造を持つ。図36は、"menu.tdat"と"mark.tdat"のシンタクス構造を示す図である。tn_blockは、1つの符号化サムネイルピクチャがストアされる領域である。1つのサムネイルピクチャのバイト長は、1つのtn_blockの大きさ以下である。1つのピクチャデータの第1バイト目は、tn_blockの第1バイト目と一致していなければならない。

menu.tdatの場合、1つのtn_blockのサイズは、menu.tdatの中のtn_block_sizeで示される。mark.tdatの場合、1つのtn_blockのサイズはmark.tdatの中のtn_block_sizeで示される。各tn_blockは、それがシンタクス中のfor-loopの中で現れるときのtn_block_idの値で区別される。menu.tidx中のtn_block_idは、menu.tidx中のref_to_tn_block_idによって参照される。mark.tidx中のtn_block_idは、mark.tidx中のref_to_tn_block_idによって参照される。

サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は、容易に高速に実行できなければならない。この理由のため、menu.tdatとmark.tdatは、ブロック構造を有する。1つのピクチャデータは、1つのtn_blockに格納される。

menu.tdat及びmark.tdatのtn_block列の中に、使用されていないtn_blockが存在してもよい。例えば、所定のサムネイルの削除をする場合、サムネイルのヘッダ情報ファイルの中にエントリされているthumbnail_indexを消去し、サムネイルのピクチャデータファイルを何も変更しなかったとき、tn_block列の中に、使用されていないtn_blockができる。

図37は、サムネイルピクチャデータが、どのようにtn_blockに格納されるか

を模式的に表した図である。図37に示すように、1つのサムネイルピクチャのバイト長は、1つのtn_blockの大きさ以下である。tn_block列の中に、使用されていないtn_blockが存在してもよい。

上述した例においては、例えば、menu.tbatとmark.tbatのファイルが2個に分けられて記録されるとして説明したが、それぞれのファイルをブロックと考え、menu.tbatのデータの第1のブロックと、mark.tbatのデータの第2のブロックを、1つのファイルにまとめて記録するようにしてもよい。1つのファイルにまとめて記録するとは、例えば、Clip Information fileの中で、SequenceInfo、C P I、ClipMarkを別のブロックとして記録するような形態にしてもよい。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば図38に示すような構成の汎用のパーソナルコンピュータ等に、記録媒体からインストールされる。

図38に示すパーソナルコンピュータにおいて、C P U (Central Processing Unit) 201は、R O M (Read Only Memory) 202に記憶されているプログラム、又は記憶部208からR A M (Random Access Memory) 203にロードされたプログラムに従って各種の処理を実行する。R A M 203にはまた、C P U 201が各種の処理を実行する上において必要なデータ等も適宜記憶される。

C P U 201、R O M 202、及びR A M 203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インタフェース205も接続されている。

入出力インタフェース205には、キーボード、マウス等よりなる入力部206、C R T、L C D等よりなるディスプレイ、並びにスピーカ等よりなる出力部207、ハードディスク等より構成される記憶部208、モデム、ターミナルアダプタ等より構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

入出力インタフェース205にはまた、必要に応じてドライブ210が接続さ

れ、磁気ディスク 221、光ディスク 222、光磁気ディスク 223、或いは半導体メモリ 224 等が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 208 にインストールされる。

この記録媒体は、図 38 に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク 221 (フロッピディスクを含む)、光ディスク 222 (CD-ROM (Compact Disk-Read Only Memory)、DVD (Digital Versatile Disk) を含む)、光磁気ディスク 223 (MD (Mini-Disk) を含む)、若しくは半導体メモリ 224 等よりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されている ROM 202 や記憶部 208 が含まれるハードディスク等で構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上の如く、本発明に係る記録装置及び方法、並びに第 1 のプログラムにおいては、入力された動画データから、当該動画データを代表するサムネイル画像を第 1 のサムネイルデータとして生成すると共に、動画データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第 2 のサムネイルデータとして生成し、第 1 のサムネイルデータと第 2 のサムネイルデータを、各々独立したグループとして記録媒体に記録するようにしたので、その記録されている画像データを用いることにより、ユーザが記録されているデータの選択を簡便に行える。

また、本発明に係る再生装置及び方法、並びに第 2 のプログラムによれば、A

Vストリームの再生が指示された場合、A Vストリームの内容を示すサムネイル画像の画像データを管理する管理データを読み出し、読出手段により読み出された管理データに基づき、画像データを読み出し、A Vストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、又は、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出し、読み出された管理データに基づき、画像データを読み出すようにしたので、その読み出された管理データにより、ユーザが記録されているデータの選択を簡便に行える。

請求の範囲

1. 入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成手段と、

前記生成手段により生成された前記第1のサムネイルデータと前記第2のサムネイルデータを、各々独立したグループとして記録媒体に記録する記録手段とを有する記録装置。

2. 前記生成手段は、前記第1のサムネイルデータと前記第2のサムネイルデータを、各々独立したファイルとして生成する請求の範囲第1項に記載の記録装置。

3. 前記生成手段は、前記第1のサムネイルデータと前記第2のサムネイルデータを各々独立したデータブロックとするとともに1つのファイルとして生成する請求の範囲第1項に記載の記録装置。

4. 前記生成手段は、前記第1のサムネイルデータに対応する番号で前記第1のサムネイルデータを管理する第1の管理データも生成すると共に、前記第2のサムネイルデータに対応する番号で前記第2のサムネイルデータを管理する第2の管理データも生成し、

前記記録手段は、前記第1の管理データ及び前記第2の管理データを前記記録媒体に記録する請求の範囲第1項に記載の記録装置。

5. 前記第1の管理データと前記第2の管理データは、管理する前記サムネイル画像の画像データのフォーマット形式を示すデータを含む請求の範囲第4項に記載の記録装置。

6. 前記記録手段は、前記第1のサムネイルデータ又は前記第2のサムネイルデータに含まれる前記サムネイル画像の画像データを、所定の大きさのブロック単位で記録する請求の範囲第4項に記載の記録装置。

7. 前記記録手段は、前記第1のサムネイル画像の参照先を示す情報を更に別

ファイルとして前記記録媒体に記録する請求の範囲第1項に記載の記録装置。

8. 前記記録手段は、前記第2のサムネイルデータに含まれる前記サムネイル画像の参照先を示す情報を更に記録する請求の範囲第1項に記載の記録装置。

9. 入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、前記生成ステップの処理で生成された前記第1のサムネイルデータと前記第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含む記録方法。

10. 入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、

前記生成ステップの処理で生成された前記第1のサムネイルデータと前記第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

11. 入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、又は、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、

前記生成ステップの処理で生成された前記第1のサムネイルデータと前記第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

12. 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データを読み出す第1の読出手段と、

前記読出手段により読み出された前記管理データに基づき、前記画像データを読み出す第2の読出手段と、

前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、又は、ユーザにより指定された画像のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出手段と、

前記第3の読出手段により読み出された前記管理データに基づき、前記画像データを読み出す第4の読出手段とを含む再生装置。

13. 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読出制御ステップと、

前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、又は、ユーザにより指定された画像のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出制御ステップと、

前記第3の読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読出制御ステップとを含む再生方法。

14. 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読出制御ステップと、

前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、又は、ユーザにより指定された画像のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出制御ステップと、

前記第3の読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読出制御ステップとを含むコンピュー

タが読み取り可能なプログラムが記録されている記録媒体。

15. 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読出制御ステップと、

前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、又は、ユーザにより指定された画像のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読出制御ステップと、

前記第3の読出制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読出制御ステップとをコンピュータに実行させるプログラム。

16. 画像データ、前記画像データから、前記画像データの内容を示すサムネイル画像の画像データと、そのサムネイル画像のデータを管理する管理データから構成される第1のデータ、及び、前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、又は、ユーザが指定した画像のサムネイル画像の画像データと、その画像データを管理する管理データから構成される第2のデータが記録されている記録媒体。

1/34

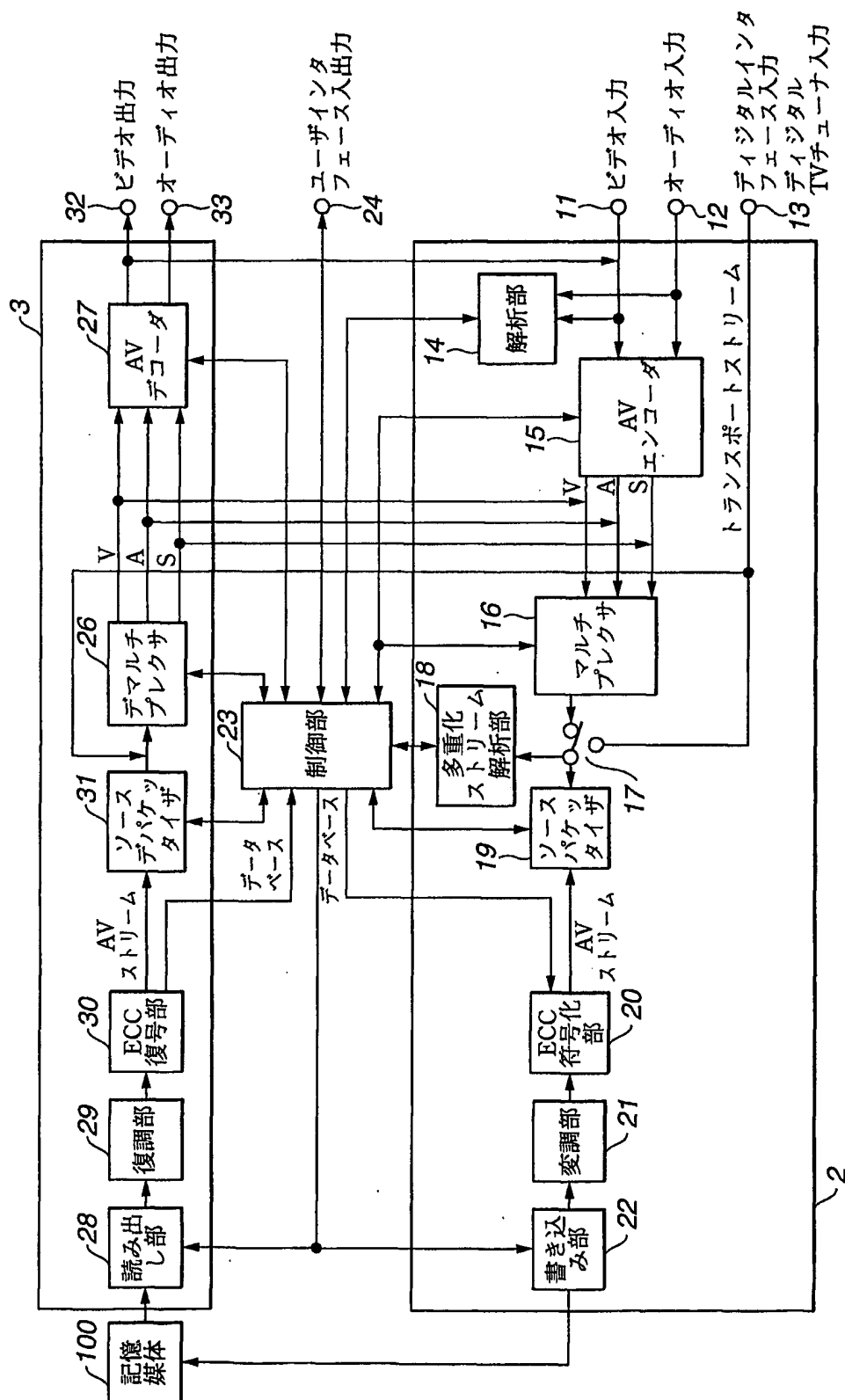


FIG.1

2/34

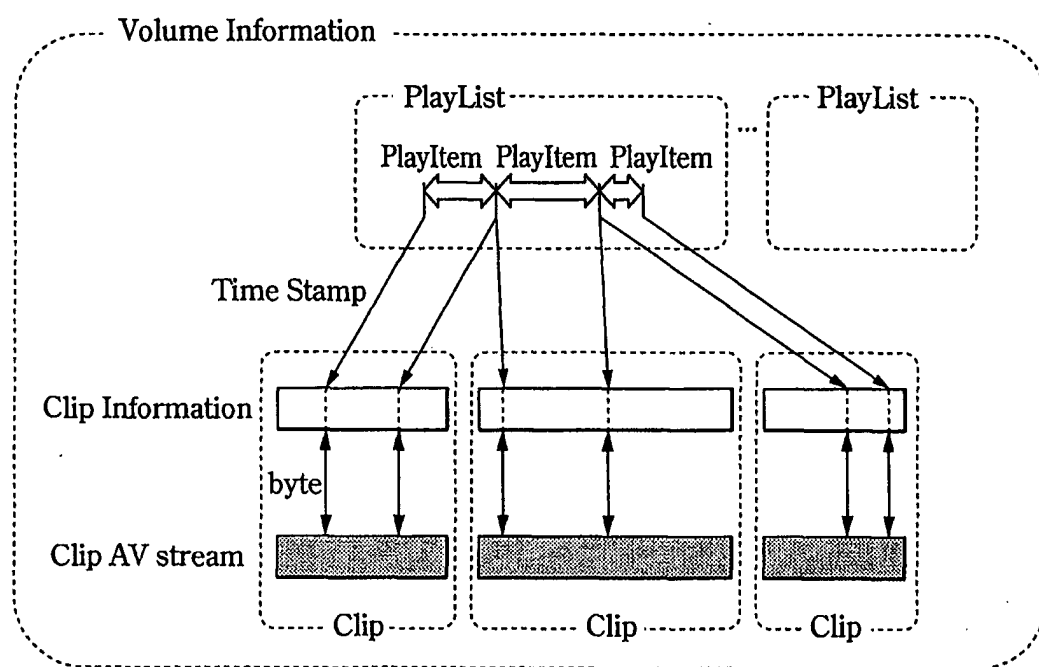


FIG.2

3/34

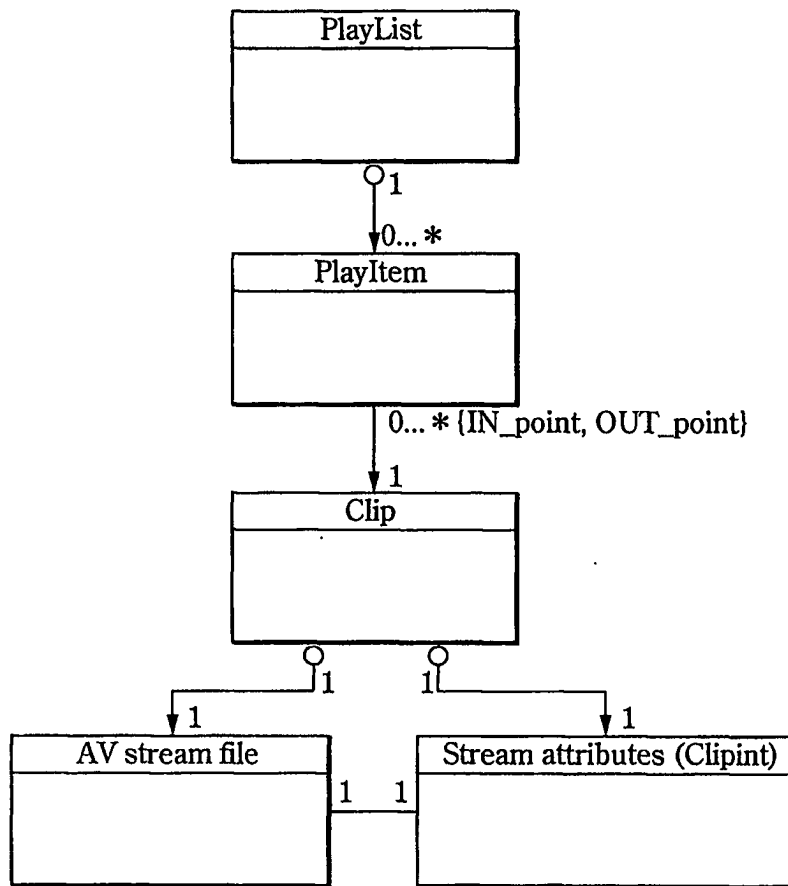
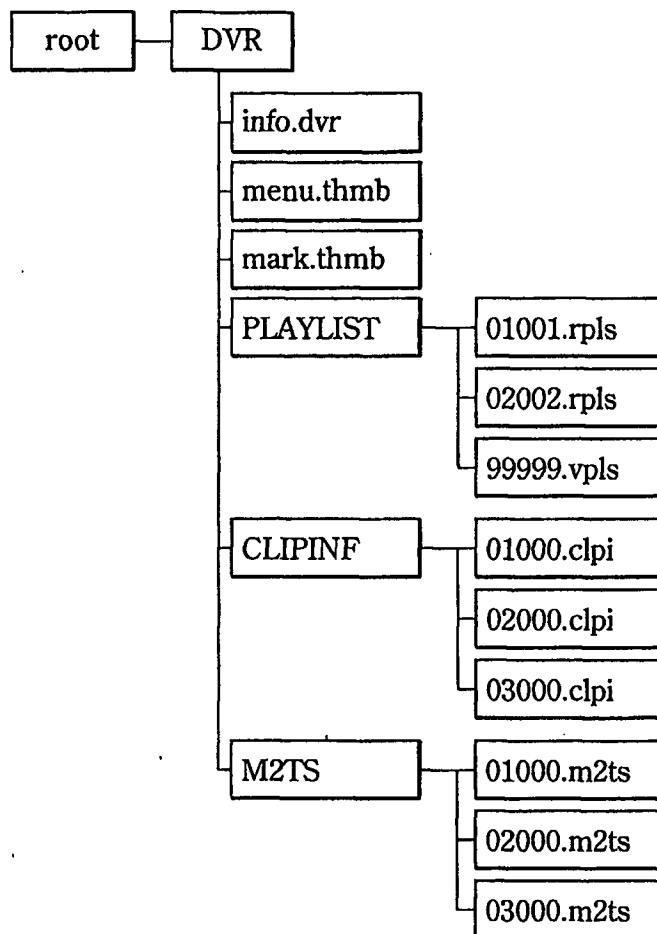


FIG.3

4/34

**FIG.4**

5/34

| シンタクス | バイト数 | 略号 |
|---------------------------------------|------|--------|
| info.dvr { | | |
| TableOfPlayLists_Start_address | 32 | uimsbf |
| MakerPrivateData_Start_address | 32 | uimsbf |
| reserved | 192 | bslbf |
| DVRVolume() | | |
| for (i=0;i<N1;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| TableOfPlayLists() | | |
| for (i=0;i<N2;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| MakerPrivateData() | | |
| } | | |

FIG.5

6/34

| シンタクス | バイト数 | 略号 |
|--------------------------|------|--------|
| DVRVolume(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| ResumeVolume() | | |
| UIAppInfoVolume() | | |
| } | | |

FIG.6

7/34

| シンタクス | バイト数 | 略号 |
|------------------------------|-------|--------|
| UIAppInfoVolume(){ | | |
| character_set | 8 | bslbf |
| name_length | 8 | uimsbf |
| Volume_name | 8*256 | bslbf |
| reserved | 15 | bslbf |
| Volume_protect_flag | 1 | bslbf |
| PIN | 8*4 | bslbf |
| ref_thumbnail_index | 16 | uimsbf |
| reserved | 7 | bslbf |
| rp_info_valid_flag | 1 | uimsbf |
| rp_ref_to_PlayList_file_name | 8*10 | bslbf |
| rp_ref_to_PlayItem_id | 16 | uimsbf |
| rp_time_stamp | 32 | uimsbf |
| } | | |

FIG.7

| シンタクス | バイト数 | 略号 |
|--|------|--------|
| TableOfPlayLists(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| number_of_PlayLists | 16 | uimsbf |
| for (i=0; i<number_of_PlayLists; i++){ | | |
| Playlist_file_name | 8*10 | bslbf |
| UIAppInfoPlaylist2() | | |
| } | | |
| } | | |

FIG.8

9/34

| シンタクス | バイト数 | 略号 |
|-------------------------|-------|--------|
| UIAppInfoPlayList20{ | | |
| character_set | 8 | bslbf |
| name_length | 8 | uimsbf |
| PlayList_name | 8*256 | bslbf |
| reserved | 8 | bslbf |
| record_time_and_date | 4*14 | bslbf |
| reserved | 8 | bslbf |
| duration | 4*6 | bslbf |
| valid_period | 4*8 | bslbf |
| maker_id | 16 | bslbf |
| maker_code | 16 | bslbf |
| reserved | 11 | bslbf |
| playback_control_flag | 1 | bslbf |
| write_protect_flag | 1 | bslbf |
| is_played_flag | 1 | bslbf |
| archive | 2 | bslbf |
| ref_thumbnail_index | 16 | uimsbf |
| reserved | 7 | bslbf |
| rp_info_valid_flag | 1 | uimsbf |
| rp_ref_to_PlayItem_id | 16 | uimsbf |
| rp_time_stamp | 32 | uimsbf |
| reserved_for_future_use | 240 | bslbf |
| } | | |

FIG.9

10/34

| シンタクス | バイト数 | 略号 |
|--------------------------------|------|--------|
| zzzzz.clpi { | | |
| STC_Info_Start_address | 32 | uimsbf |
| ProgramInfo_Start_address | 32 | uimsbf |
| CPI_Start_address | 32 | uimsbf |
| ClipMark_Start_address | 32 | uimsbf |
| MakerPrivateData_Start_address | 32 | uimsbf |
| reserved | 96 | bslbf |
| ClipInfo() | | |
| for (i=0;i<N1;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| STC_Info() | | |
| for (i=0;i<N2;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| ProgramInfo() | | |
| for (i=0;i<N3;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| CPI() | | |
| for (i=0;i<N4;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| ClipMark() | | |
| for (i=0;i<N5;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| MakerPrivateData() | | |
| } | | |

FIG.10

11/34

| シンタクス | バイト数 | 略号 |
|---|-------|--------|
| ClipMark(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| number_of_Clip_marks | 16 | uimsbf |
| for (i=0; i<number_of_Clip_marks; i++){ | | |
| reserved | 8 | bslbf |
| mark_type | 8 | bslbf |
| mark_time_stamp | 32 | uimsbf |
| STC_sequence_id | 8 | uimsbf |
| reserved | 24 | bslbf |
| character_set | 8 | bslbf |
| name_length | 8 | uimsbf |
| mark_name | 8*256 | bslbf |
| ref_thumbnail_index | 16 | uimsbf |
| } | | |
| } | | |

FIG.11

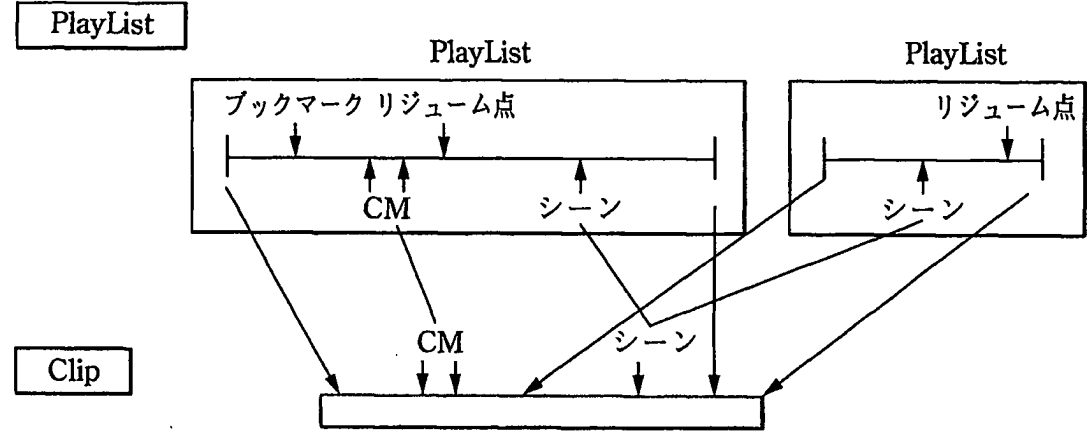


FIG.12

13/34

| シンタクス | バイト数 | 略号 |
|---------------------------------------|------|--------|
| xxxxxx.rpls / yyyyyy.vpls { | | |
| PlayListMark_Start_address | 32 | uimsbf |
| MakerPrivateData_Start_address | 32 | uimsbf |
| reserved | 192 | bslbf |
| PlayList() | | |
| for (i=0;i<N1;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| PlayListMark() | | |
| for (i=0;i<N2;i++){ | | |
| padding_word | 16 | bslbf |
| } | | |
| MakerPrivateData() | | |
| } | | |

FIG.13

14/34

| シンタクス | バイト数 | 略号 |
|---|------|--------|
| PlayList(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| PlayList_type | 8 | uimsbf |
| CPI_type | 1 | bslbf |
| reserved | 7 | bslbf |
| UIAppInfoPlayList() | | |
| number_of_PlayItems // main path | 16 | uimsbf |
| if (<Virtual PlayList>){ | | |
| number_of_SubPlayItems // sub path | 16 | uimsbf |
| }else{ | | |
| reserved | 16 | bslbf |
| } | | |
| for (PlayItem_id=0; | | |
| PlayItem_id<number_of_PlayItems; | | |
| PlayItem_id++){ | | |
| PlayItem() //main path | | |
| } | | |
| if (<Virtual PlayList>){ | | |
| if (CPI_type==0 && PlayList_type==0){ | | |
| for (i=0; i<number_of_SubPlayItems; i++) | | |
| SubPlayItem() //sub path | | |
| } | | |
| } | | |
| } | | |

FIG.14

| シンタクス | バイト数 | 略号 |
|----------------------------------|------|--------|
| PlayItem(){ | | |
| Clip_information_file_name | 8*10 | bslbf |
| reserved | 24 | bslbf |
| STC_sequence_id | 8 | uimsbf |
| IN_time | 32 | uimsbf |
| OUT_time | 32 | uimsbf |
| reserved | 14 | bslbf |
| connection_condition | 2 | bslbf |
| if (<Virtual PlayList>){ | | |
| if (connection_condition=='10'){ | | |
| BridgeSequenceInfo() | | |
| } | | |
| } | | |
| } | | |

FIG.15

16/34

| シンタクス | バイト数 | 略号 |
|---|-------|--------|
| PlayListMark(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| number_of_PlayList_marks | 16 | uimsbf |
| for (i=0;i<number_of_PlayList_marks;i++){ | | |
| reserved | 8 | bslbf |
| mark_type | 8 | bslbf |
| mark_time_stamp | 32 | uimsbf |
| PlayItem_id | 8 | uimsbf |
| reserved | 24 | uimsbf |
| character_set | 8 | bslbf |
| name_length | 8 | uimsbf |
| mark_name | 8*256 | bslbf |
| ref_thumbnail_index | 16 | uimsbf |
| } | | |
| } | | |

FIG.16

17/34

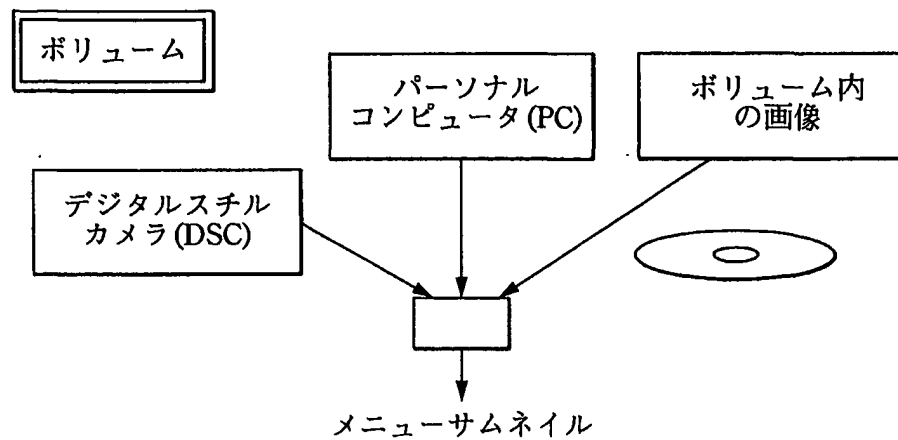


FIG.17

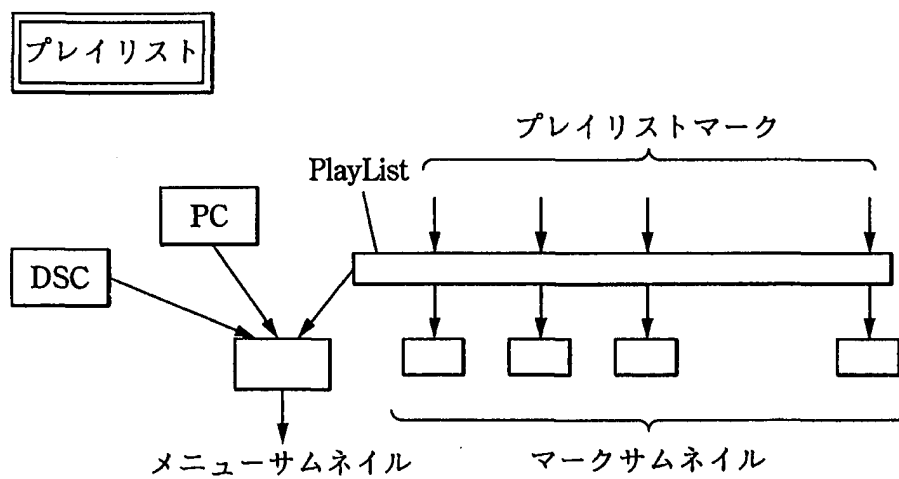


FIG.18

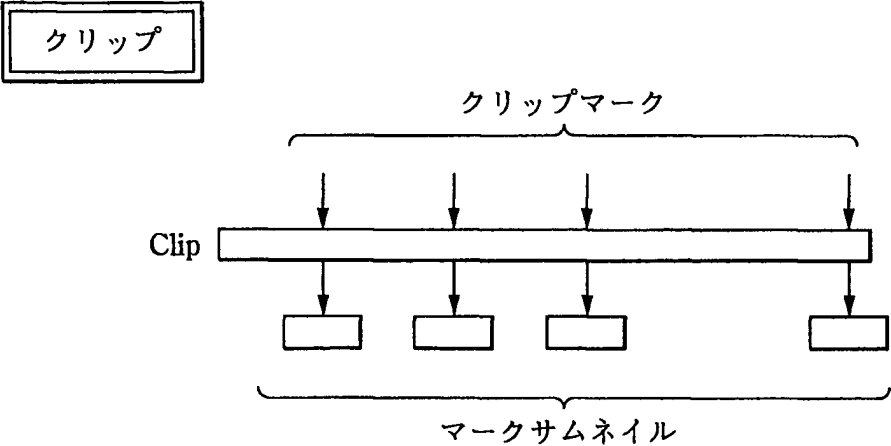


FIG.19

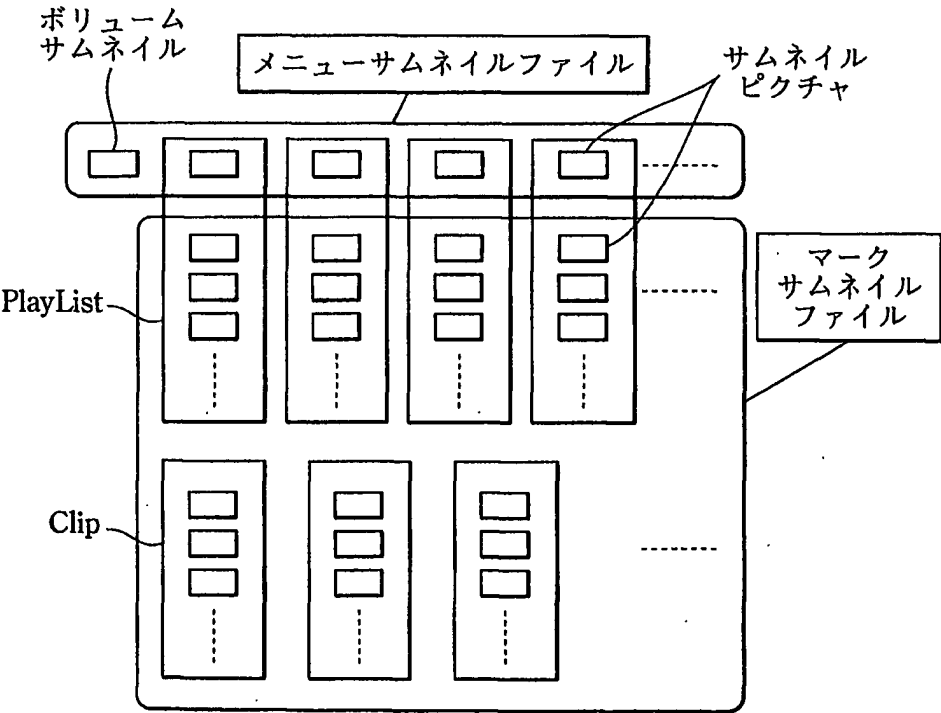


FIG.20

| シンタクス | バイト数 | 略号 |
|-----------------------|------|-------|
| menu.thmb/mark.thmb { | | |
| reserved | 256 | bslbf |
| Thumbnail() | | |
| for (i=0;i<N1;i++) | | |
| padding_word | 16 | bslbf |
| } | | |

FIG.21

20/34

| シンタクス | バイト数 | 略号 |
|---|--------|--------|
| Thumbnail(){ | | |
| version_number | 8*4 | char |
| length | 32 | uimsbf |
| if (length !=0){ | | |
| tn_blocks_start_address | 32 | bslbf |
| number_of_thumbnails | 16 | uimsbf |
| tn_block_size | 16 | uimsbf |
| number_of_tn_blocks | 16 | uimsbf |
| reserved | 16 | bslbf |
| for (i=0; i<number_of_thumbnails; i++){ | | |
| thumbnail_index | 16 | uimsbf |
| thumbnail_picture_format | 8 | bslbf |
| reserved | 8 | bslbf |
| picture_data_size | 32 | uimsbf |
| start_tn_block_number | 16 | uimsbf |
| x_picture_length | 16 | uimsbf |
| y_picture_length | 16 | uimsbf |
| reserved | 16 | uimsbf |
| } | | |
| stuffing_bytes | 8*2*L1 | bslbf |
| for(k=0; k<number_of_tn_blocks; k++){ | | |
| tn_block | fixed | |
| } | | |
| } | | |

FIG.22

| Thumbnail_picture_format | 意味 |
|--------------------------|------------------------|
| 0x00 | MPEG-2 Video I-picture |
| 0x01 | DCF (restricted JPEG) |
| 0x02 | PNG |
| 0x03-0xff | reserved |

FIG.23

FIG.24A

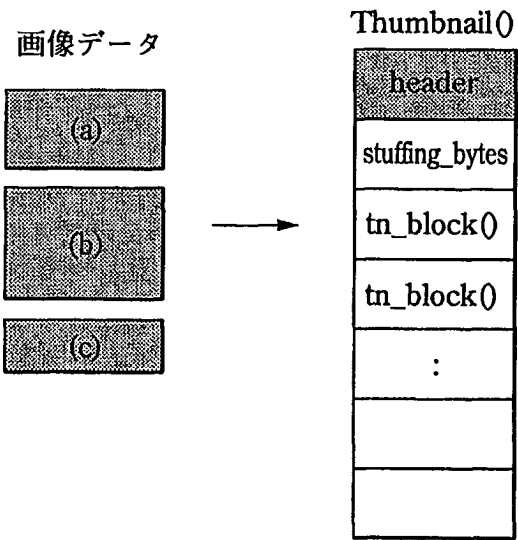
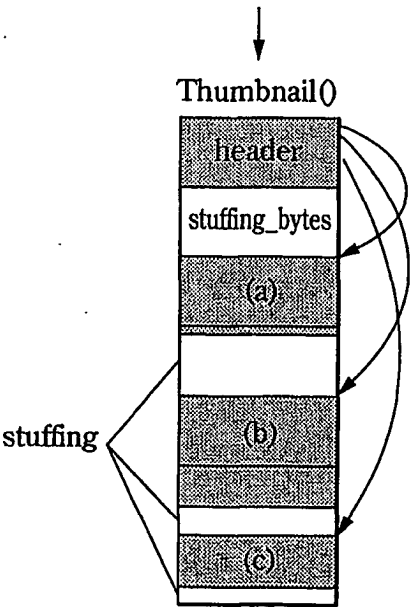


FIG.24B



22/34

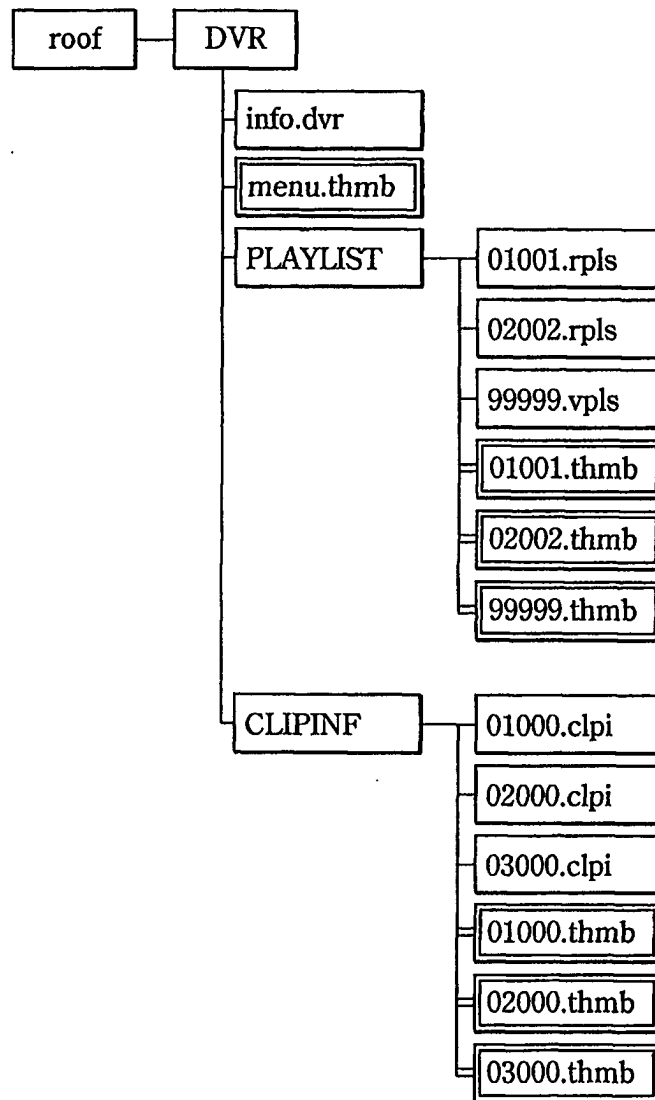


FIG.25

| シンタクス | バイト数 | 略号 |
|------------------------|------|-------|
| menu.thmb/xxxxx.thmb { | | |
| reserved | 256 | bslbf |
| Thumbnail() | | |
| for (i=0;i<N1;i++) | | |
| padding_word | 16 | bslbf |
| } | | |

FIG.26

24/34

| シンタクス | バイト数 | 略号 |
|--|------|--------|
| TableOfPlayLists{ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| number_of_PlayLists | 16 | uimsbf |
| for (i=0; i<number_of_PlayLists; i++){ | | |
| PlayList_file_name | 8*10 | bslbf |
| } | | |
| } | | |

FIG.27

25/34

| シンタクス | バイト数 | 略号 |
|---|------|--------|
| PlayList(){ | | |
| version_number | 8*4 | bslbf |
| length | 32 | uimsbf |
| PlayList_type | 8 | uimsbf |
| CPI_type | 1 | bslbf |
| reserved | 7 | bslbf |
| UIAppInfoPlayList() | | |
| number_of_PlayItems // main path | 16 | uimsbf |
| if (<Virtual PlayList>){ | | |
| number_of_SubPlayItems // sub path | 16 | uimsbf |
| }else{ | | |
| reserved | 16 | bslbf |
| } | | |
| for (PlayItem_id=0; | | |
| PlayItem_id<number_of_PlayItems; | | |
| PlayItem_id++){ | | |
| PlayItem() //main path | | |
| } | | |
| if (<Virtual PlayList>){ | | |
| if (CPI_type==0 && PlayList_type==0){ | | |
| for (i=0; i<number_of_SubPlayItems; i++) | | |
| SubPlayItem() //sub path | | |
| } | | |
| } | | |
| } | | |

FIG.28

| シンタクス | バイト数 | 略号 |
|-------------------------|-------|--------|
| UIAppInfoPlayList(){ | | |
| character_set | 8 | bslbf |
| name_length | 8 | uimsbf |
| PlayList_name | 8*256 | bslbf |
| reserved | 8 | bslbf |
| record_time_and_date | 4*14 | bslbf |
| reserved | 8 | bslbf |
| duration | 4*6 | bslbf |
| valid_period | 4*8 | bslbf |
| maker_id | 16 | uimsbf |
| maker_code | 16 | uimsbf |
| reserved | 11 | bslbf |
| playback_control_flag | 1 | bslbf |
| write_protect_flag | 1 | bslbf |
| is_played_flag | 1 | bslbf |
| archive | 2 | bslbf |
| ref_thumbnail_index | 16 | uimsbf |
| reserved_for_future_use | 240 | bslbf |
| } | | |

FIG.29

27/34

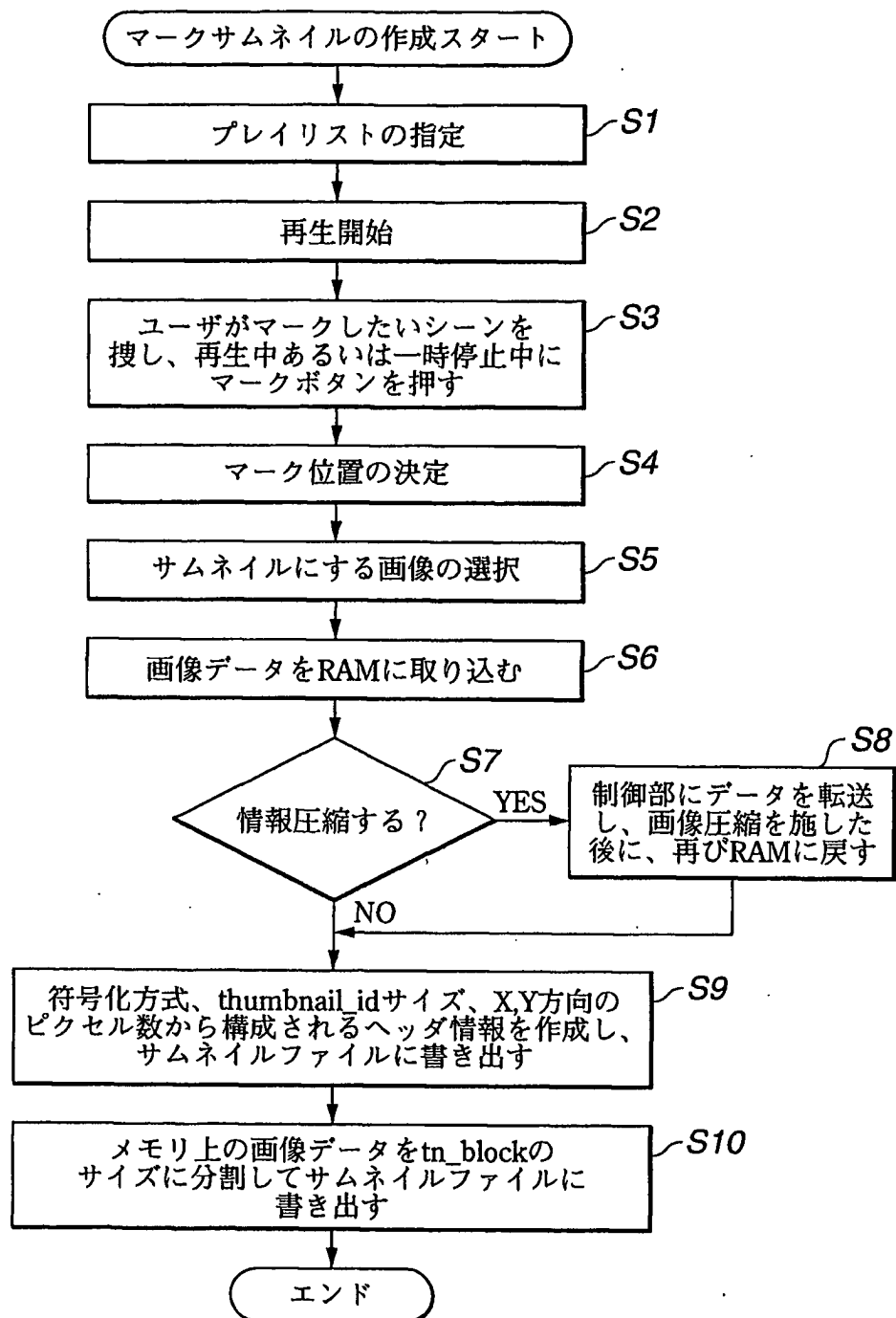


FIG.30

28/34

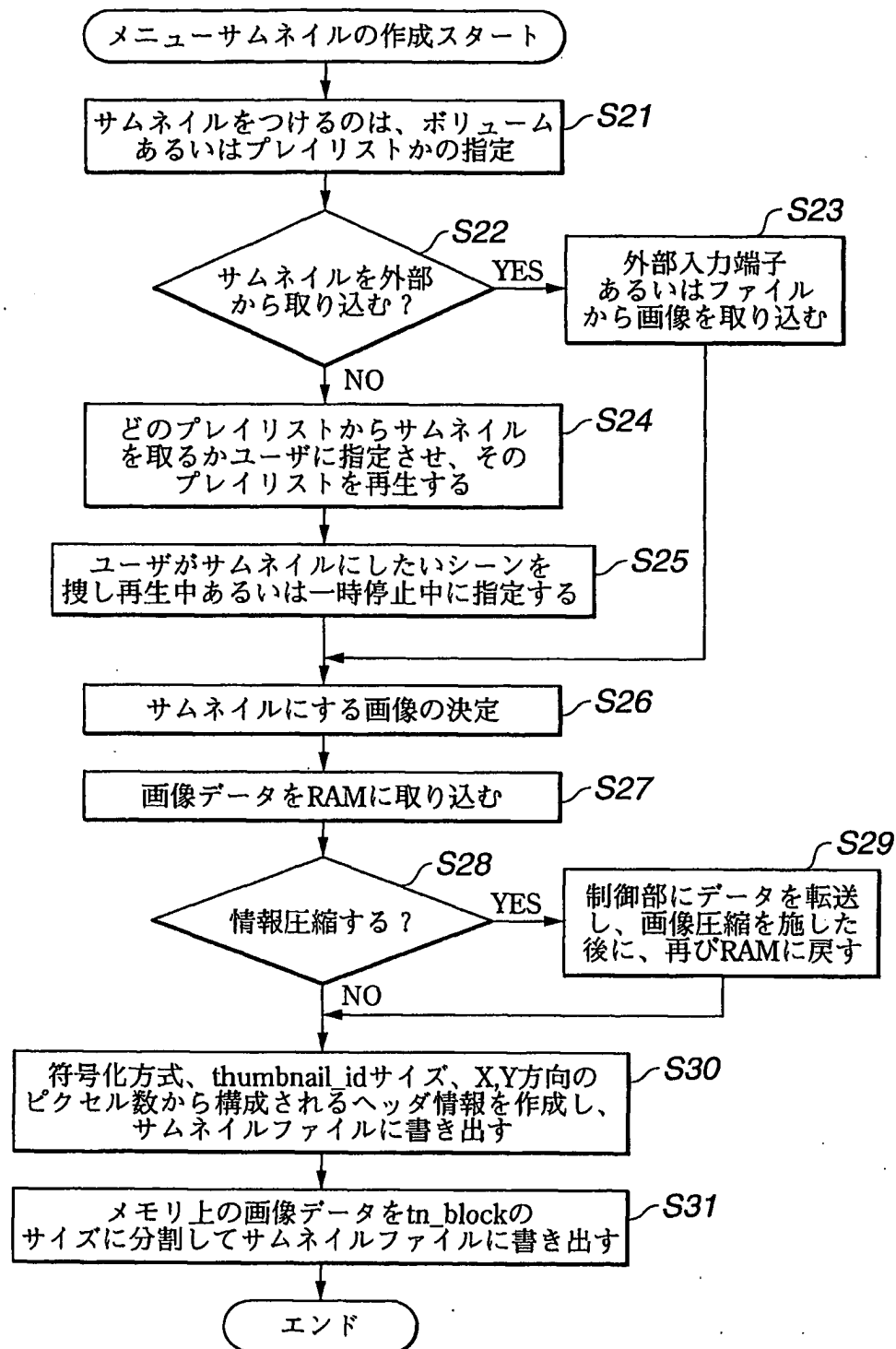


FIG.31

29/34

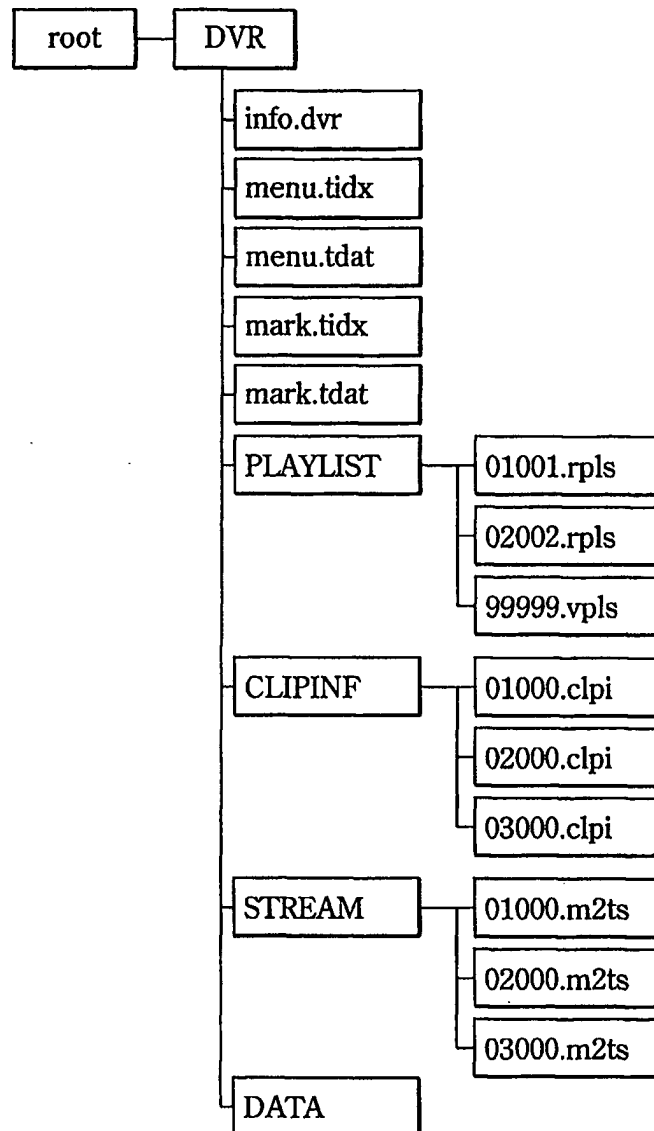


FIG.32

30/34

| シンタクス | バイト数 | 略号 |
|---|------|--------|
| menu.tidx/mark.tidx){ | | |
| version_number | 8*4 | char |
| reserved_for_future_use | 256 | bslbf |
| length | 32 | uimsbf |
| if (length !=0){ | | |
| number_of_thumbnails | 16 | uimsbf |
| tn_block_size | 16 | uimsbf |
| number_of_tn_blocks | 16 | uimsbf |
| for (i=0; i<number_of_thumbnails; i++){ | | |
| thumbnail_index | 16 | uimsbf |
| ref_to_tn_block_id | 16 | uimsbf |
| picture_byte_size | 32 | uimsbf |
| horizontal_picture_size | 16 | uimsbf |
| vertical_picture_size | 16 | uimsbf |
| display_aspect_ratio | 4 | uimsbf |
| color_space | 4 | uimsbf |
| reserved_for_word_align | 8 | bslbf |
| } | | |
| } | | |
| } | | |

FIG.33

31/34

| display_aspect_ratio | 意味 |
|----------------------|---------------------------|
| 0, 1 | reserved for future use |
| 2 | 4:3 display aspect ratio |
| 3 | 16:9 display aspect ratio |
| 4-15 | reserved for future use |

FIG.34

| display_aspect_ratio | 意味 |
|----------------------|-------------------------|
| 0 | BT 709 |
| 1 | ITU-R Rec.601 |
| 2-15 | reserved for future use |

FIG.35

| シンタクス | バイト数 | 略号 |
|---|--------------------------|----|
| menu.tdat/mark.tdat { | | |
| for (tn_block_id=0; tn_block_id<number_of_tn_blocks; tn_block_id++){ | | |
| tn_block | tn_block_ size*1024*8 | |
| } | | |
| } | | |

FIG.36

33/34

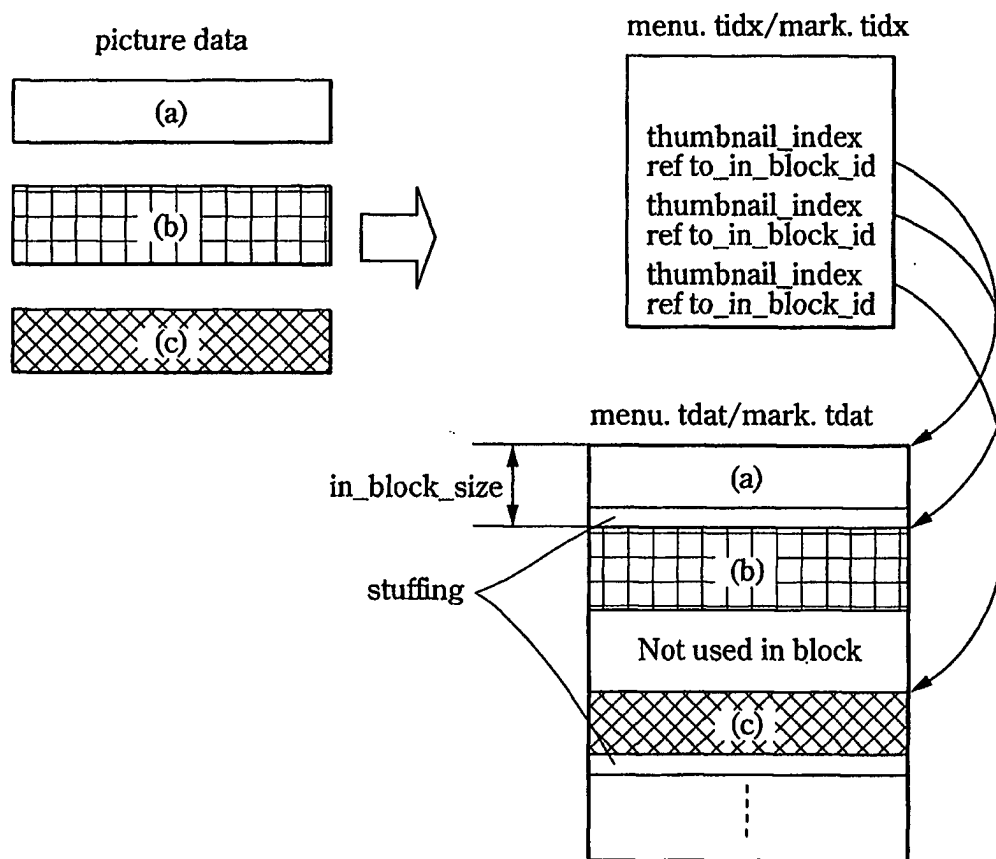


FIG.37

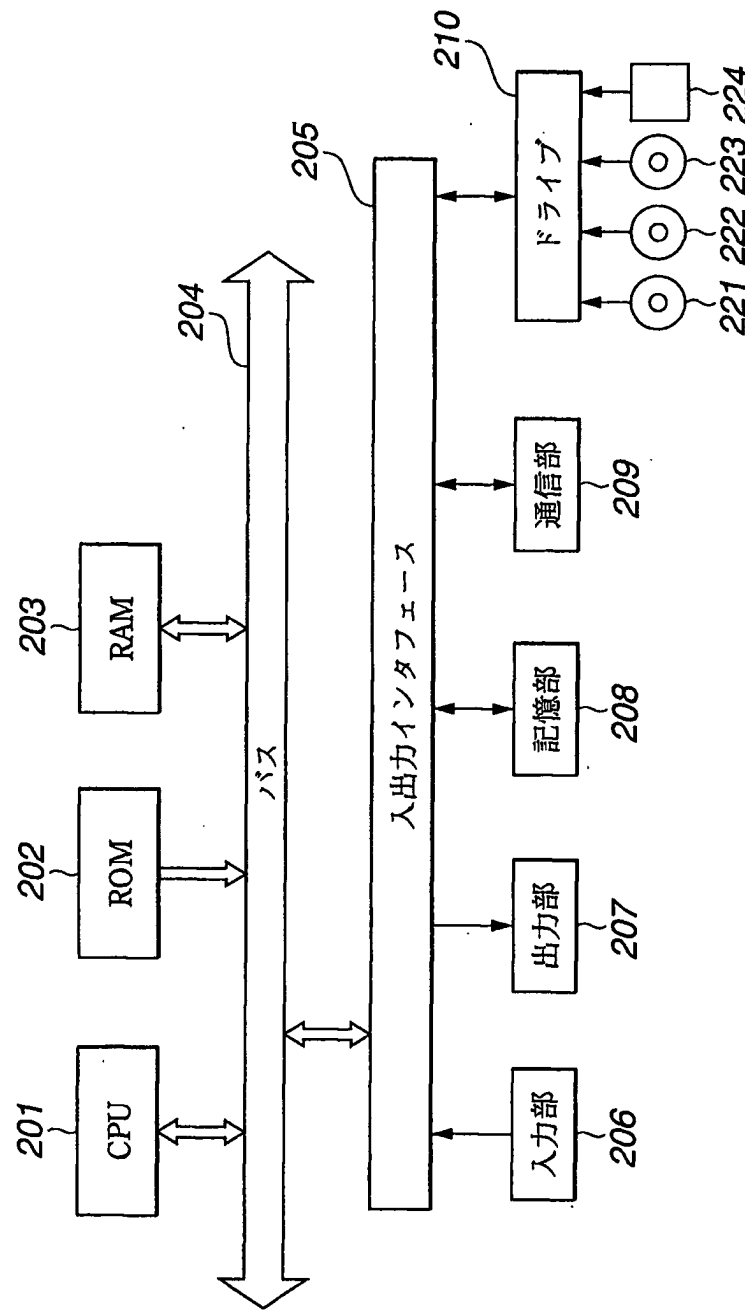


FIG.38